I. J. Computer Network and Information Security, 2012, 5, 63-71
Published Online June 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.05.08

An Analytical Assessment on Document Clustering

Pushplata, Mr. Ram Chatterjee
Maharishi Dayanand University, Rohtak, Manva Rachna Collage of Engineering, Faridabad, India
pushp12lata@gmail.com, ramchatterjee.mrce@mrei.ac.in

Abstract— Clustering is related to data mining for information retrieval. Relevant information is retrieved quickly while doing the clustering of documents. It organizes the documents into groups; each group contains the documents of similar type content. Document clustering is an unsupervised approach of data mining. Different clustering algorithms are used for clustering the documents such as partitioned clustering (K-means Clustering) and Hierarchical Clustering (Agglomerative Hierarchical Clustering (AHC)). This paper presents analysis of Suffix Tree Clustering (STC) Algorithm and other clustering techniques (K-means, AHC) that are being done in literature survey. The paper also focuses on traditional Vector Space Model (VSM) for similarity measures, which is used for clustering the documents. This paper also focuses on the comparison of different clustering algorithms. STC algorithm improves the searching performance as compare to other clustering algorithms as the papers studied in literature survey. The paper presents STC algorithm applied on the search result documents, which is stored in the dataset. This paper articulates the key requirements for web document clustering and clusters would be created on the full text of the web documents. STC perform the clustering and make the clusters based on phrases shared between the documents. STC is faster clustering algorithm for document clustering.

Index Terms— Data mining, Document clustering, Suffix Tree Clustering (STC) steps, K-means, Agglomerative Hierarchical Clustering (AHC), cosine similarity

I. Introduction

Clustering is raised from the Data Mining. Data mining have methods such as Classification, Regression, Clustering and summarization [2]. Data mining is used to retrieve the information from large repository of data and extraction of hidden information from large databases.

Data mining tools predict the future trends and behavior. Web mining is used to perform the mining for the web documents. The paper focuses on both data mining and web mining techniques for information retrieval.

Due to explosive growth of accessing information from the web, efficient access and exploration of information are needed critically. Most of the search engine returns the long list of documents partial content matching the search query. While doing searching, there is many irrelevant documents are returned with the relevant documents. Clustering helps us to speed up the knowledge discovery. Searching on the web is tedious and time consuming, to reduce this time clustering algorithms would be used. The key requirements [12] for web document clustering are:

Relevance: Relevancy defines that, the documents which is related to user query are relevant that is separated from the irrelevant documents.

Browsable summaries: The content of the clusters are according to user interest. It replaces the rank list representation with the shifting through clustering. This method provides concise and accurate description of the clusters.

Overlap: In a document there are multiple topics. It is used to avoid confing of each document to only one cluster.

Snippet-tolerance: This method provides the high quality clusters when accessing the snippets returned by search engine when the users don't want to wait for the system to download the documents of the web.

Speed: Clustering allows the users to access the documents in few seconds. Clustering is process to speed up the searching while doing the browsing on the web.

Incrementally: The method should start to process each snippet as soon as it receives from the web, to save the time.

Many clustering algorithms apply on the collection of documents, which rely on off-line. But the clustering performed on small set of documents, in response to user query. The objective of clustering is to grouping the documents which contains similar type contents. The clustering is based on the four concepts: data representation model, similarity measures, clustering model and clustering algorithm.

Document clustering [3, 4] comes from the domain of the information retrieval. Document clustering is still a developing field which is undergoing evolution It finds the grouping for set of documents so that documents belong to the same cluster are similar and documents belong to different clusters are dissimilar. Document clustering is a method of automatically

organize the large data collection into groups. Document clustering treated a document as a bag of words and clustering criteria is based on the presence of similar words in document. Document clustering has always been used to improve the performance of retrieval from large data collection.

Most of the document clustering techniques [3, 4, 5] based on the concept vector space model. The documents in the dataset are represented in the vector form. The term weights are usually referred to term frequency and inverse term frequency [12] of words are contained in vector. There is another concept known as similarity measures, which is used to measures the similarity between the documents. The similarity between two documents is measured on the basis of Binary similarity, Euclidean distance measures, Manhattan distance measures, and cosine similarity measures etc. In a document phrases have been used for more accurate and effective document clustering. A phrase in a document is an ordered sequence of words. The clustering method categorized as partitioned clustering methods and hierarchical clustering methods.

Partitioned clustering [5, 7, 8, and 11]**:** Partitioned clustering algorithm partitioned the documents in to k number of clusters. Example of partitioned clustering is k-means clustering.

Hierarchical clustering [5, 7, 8, and 11]: In hierarchical clustering, the clusters of documents are arranged in tree like structure. Hierarchical clustering can be divided into Agglomerative hierarchical clustering and divisive clustering.

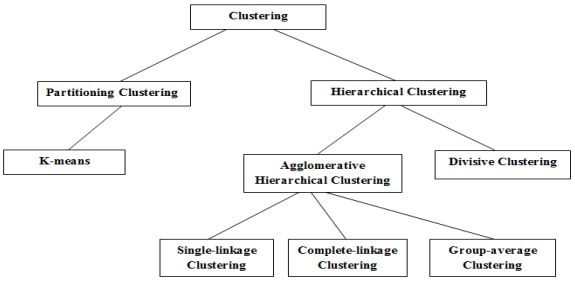

Figure 1: Different Document Clustering Techniques

II. RELATED WORK

This section reviews the previous work that is to be done on the document clustering and document clustering algorithms. Document clustering is a method, which is used for information retrieval from the text documents (data mining) and web documents (web mining). Document clustering is improved as according to the requirements of the user. Document clustering is investigated to improve the performance of the search engine [15] by pre-clustering and clustering has also been used for post-retrieval document browsing.

Due to explosive growth of accessing information from the web many researches has been done according to the requirements of the users. These researches make the information inquiry better and more convenient for the users. Many researches has been done in the field of information retrieval, data mining and web mining that is related to document clustering.

Information retrieval [2]: Large number of files are uploaded and downloaded on the internet. According to study 90% of the information is extracted from the internet. Information retrieval plays an important role in data mining and web mining. Information retrieval analyzes the file content and identifies their similarity. Information retrieval measures the performance in the documents. The performance measuring methods such as Precision and Recall describe the relevancy of the documents in information retrieval.

Document clustering [3, 4, 5, 12, and 20]: Clustering method is conducted in fully unsupervised learning. It is more flexible and adaptive method. Document clustering [14] related to the grouping of the documents on the basis of their similarity. Documents are divided in to different groups, documents in the same group means that they are similar to each other than the other documents. Clustering is absolutely helpful in speeding up the knowledge discovery. Document clustering follows the steps to make the cluster that display in the figure:

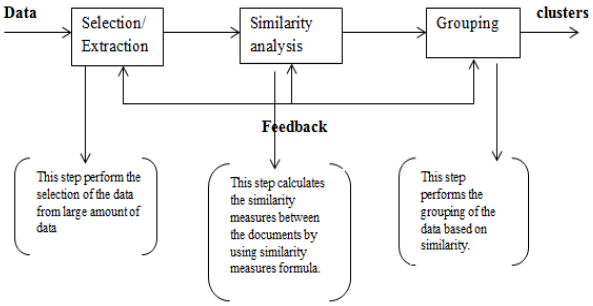

Figure 2: Clustering Process

Clustering algorithm [3, 20]: There are other clustering algorithms which are used for document clustering, which appear in the literature survey to produces the effective search results. There are two main categories of the document clustering algorithm that is Partitioned Clustering [6, 7, 8] and Hierarchical Clustering. Hierarchical clustering [10, 11] further divided into Agglomerative Hierarchical Clustering (AHC) and Divisive Hierarchical Clustering.

Hierarchical clustering initially takes each document as cluster and calculates their similarity measures. The calculation for all pair of documents demanded lots of computation. The AHC algorithm merges the two closest pair iteratively to produces the desired number of clusters. The algorithms merge the clusters like bottom-up approach for Agglomerative Hierarchical Clustering and top-down approach for Divisive Hierarchical Clustering. AHC is most commonly used for document clustering, when it is applied on the large collection of documents but it will slow down the

process. AHC further divided in to Single-link, group-link and complete-link methods.

Another clustering approach is the partitioned clustering algorithm, which creates one level partitioning of the document collection. K-means is an example of partitioned clustering algorithm. It initially selects the k documents that work as centroid of each cluster. In the next step, every document is assigned to a nearest cluster based on similarity measures. These steps were repeated until there is no change in the centroid values. Bisecting K-means is modified version of the K-means clustering algorithm.

Mostly, the Agglomerative Hierarchical Clustering [10, 11] and K-means clustering [6, 7] cannot produce the effective results for clustering the documents. Some of the clustering algorithm produces the descriptive summaries that are not meaningful and readable which is difficult and reduces the search range. To overcome the difficulties of the agglomerative hierarchical clustering and k-means clustering, A new clustering algorithm is introduced by Zamir and Etizioni,"Web Document Clustering: A feasibility demonstration" [12] which are known as Suffix Tree Clustering (STC) algorithm. STC is a search result clustering algorithm which is applied on the document collections.

STC have some advantages [14] over the other clustering algorithm such as: there is no requirement to specify the number of clusters, shared phrases describe the resultant clusters, and single document may appear in more than one cluster. STC has readable labels and descriptive summaries for resultant clusters.

III.CLUSTERING ALGORITHMS

There are different clustering algorithms which are used for clustering the documents that are described as follows:

1. Partitioned Clustering Algorithm[3, 6, 7]
2. Hierarchical Clustering Algorithm [10, 11]

1. Partitioned Clustering Algorithm [3, 20]

In partitioned clustering data is divided into partitions. There are n data objects that are divided into k number of clusters/partitions.

It uses an iterative relocation technique to improve the partitioning criteria and to find optimal partition. K-means clustering algorithm is an example of partitioned clustering algorithm.

A. K-Means clustering algorithm [6, 7, 8, 9, and 20]:

As the researchers describe that clustering is the process of partitioning the given set of objects into disjoint subsets. This is done in such a way objects in the same cluster are similar, and the objects in the different clusters are dissimilar with respect to their attributes.

K-means clustering is produces an effective result while producing the clusters. Many of the researcher's work on improve the performance and efficiency of k-means clustering algorithm. K-means clustering is one of the most simple and non-supervised clustering algorithm. For large data collection the computational complexity of the original k-means is very high. The algorithm produces the results in different clusters depending on the choice of the randomly selected initial centroids.

This section describes the k-means clustering algorithm. The algorithm classify dataset in to k disjoint clusters, where the value if the k is fixed initially. K-means clustering is based on the centroid value that's why it is called the centroid based technique. The algorithm performs in two phases: in the first phase, randomly select the k centers, the value of k is fixed for each of the cluster. In the next phase, each point belonging to given dataset and assign to its nearest centroid. Euclidean distance can be used to measure the distance between the data points and the centroid values. When a cluster contains all the data points, then the first phase is completed. At that point, we need to recalculate the centroid value and change the cluster centroid. New data points may lead to the centroid values. Once we find the k new centroid, a new binding is to be created on between some data points and nearest new centroid, generating a loop. As a result of loop k centroid change their position in step by step. The similarity between the clusters is measured by calculating the means values of data objects (data points) in a cluster. The mean value in every iteration will be calculated, while making the clusters. K is positive integer number.

Steps for K-means clustering [7] *algorithm:*

Input:

D = {d1, d2, d3,……., dn} // set of n data items.

k // desired number of clusters.

Output:

A set of k clusters.

Steps:

1. For initial centroid, select the k data items from documents D ;
2. Repeat
   Allocate each item d1 to the cluster which has the closest centroid;
   New mean values are calculated for each cluster;

Until specified condition criteria is met.

The mean value is calculated on the basis of the formula such as Euclidean distance and Manhattan distance that is defined as:

In *Euclidean distance* the distance is measured between two points such as X (x1, x2) and Y (y1, y2).

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

In *Manhattan distance* the distance is measured between two pair of objects are:

$$d(i, j) = |x_{i1}-x_{j1}| + |x_{i2}-x_{j2}| + \ldots\ldots\ldots + |x_{in}-x_{jn}| \quad (2)$$

When the Euclidean distance and Manhattan distance are used for measuring the distance the following properties are satisfied:

d (i, j)>=0: distance is a non-negative number.

d (i, i) =0: The distance of an object to itself is 0.

d (i, j) = d (j, i): distance is a symmetric function.

The number of data objects is less than the number of cluster than assign the data as a centroid of the cluster. Each centroid will have a cluster number. If the number of data objects is bigger than the number of cluster, calculate the distance to all centroid and minimum distance. The location of the centroid is based on the current updated data; assign all data to the new centroid. This process is repeated till that no data is moving to another cluster any more.

K-mean clustering used to minimize the squared error function. This method is relatively scalable and efficient in the processing of large data sets. The complexity of the algorithm is O (nkt). Where n is the total number of objects, k is the number of clusters and t is the number of iterations.

The k-means clustering algorithm is extensively clustering algorithm and it generally produces the effectively good results. The main drawback of this algorithm is that it produces the different clusters for different sets of values of the initial centroid. The quality of the clusters depends on the choice of the randomly selected initial centroid. The k-means algorithm is computationally expensive.

Properties of k-mean clustering algorithm:

- There are always k clusters. Necessity for users to specify k clusters.
- There is always at least one item in each cluster.
- The clusters are non-hierarchical and they do-not overlap.

Advantages

- If there is large number of variables k-means clustering is computationally faster than hierarchical clustering.
- K-mean produces tighter clusters than hierarchical clustering.

Disadvantages

- There is necessity for specifying K number of clusters. Fixed number of clusters can make it difficult to predict what K should be.
- Difficulty in comparing the quality of clusters.

2. Hierarchical Clustering Algorithm [19]

Search result clustering help users to quickly browse the documents returned by search engine. Hierarchical clustering is one of another method for clustering the documents. It produces the hierarchical structure of the documents.

Hierarchical methods are grouped the data objects in to tree of clusters. It output a hierarchy, a structure that is more informative than the unstructured set of clusters than the partitioned clustering. Hierarchical method uses hierarchical decomposition of a given set of data objects. The advantage of hierarchical clustering comes at the cost of lower efficiency. Hierarchical document clustering [16] is better than the partitioned clustering; its main work is to build the hierarchical structure in tree of clusters whose leaf node represents the subset of document collection.

Hierarchical method can be categorized into:

- Agglomerative Hierarchical Clustering and,
- Divisive Hierarchical Clustering.

A. Agglomerative Hierarchical Clustering:

Agglomerative hierarchical clustering is bottom-up strategy; it starts with each object a separate cluster itself and merges the clusters according to distance measures. Clusters are merges until all the objects in to a single cluster till the termination condition are satisfied. It merges the clusters iteratively. Most of the hierarchical clustering method belongs to this clustering category. Hierarchical agglomerative clustering is represented by the Genograms. It is tree like structure show the relationship between objects. In Dendogram each merge is represented by the horizontal line. The y-coordinates of horizontal line are the similarity of two clusters that were merged and documents can be viewed as single cluster. The similarity measures can be calculated by-

- Single-linkage clustering
- Complete-linkage clustering
- Group-average clustering

Single-link clustering [12]

In single-linkage clustering, the similarity between two clusters is based on the most similar members of the clusters. . In other words we can say that the similarity between two clusters is measured by the similarity of the closest pair of data points belonging to different clusters. The process of merging cluster is repeat till that all objects merged to form a single cluster. In this clustering the minimum distance is calculated between the documents.

$$\min\{ d(x, y) : x \in \mathcal{A}, y \in \mathcal{B} \}. \quad (3)$$

Complete-linkage clustering

In complete- linkage clustering, the similarity between clusters is the similarity of their most dissimilar members. The maximum distance is calculated between the documents.

$$\max\{ d(x, y) : x \in \mathcal{A}, y \in \mathcal{B} \}. \quad (4)$$

Average-linkage clustering

Average evaluates the cluster quality based on all similarity between the documents. The mean distance is calculated between the documents.

$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y). \quad (5)$$

B. Divisive Hierarchical Clustering

It is a top-down strategy just reverse of the agglomerative hierarchical clustering. It starts with all objects in to a single cluster. A cluster is split into smaller clusters, until each object in a single cluster holds the termination condition. The divisive approach divide the data objects into disjoint groups at every step this process will continue until all objects' fall into its own cluster. The divisive hierarchical clustering, which does the reverse by starting with all objects in one cluster and subdividing them into smaller pieces. The process of clustering is based on the similarity measures. Divisive methods are not generally available, and rarely have been applied.

Algorithm steps for hierarchical clustering:

There are set of N items to be clustered and N*N distance (similarity) matrix and the basic process of hierarchical clustering is this:

Step1: Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distance between the clusters the same as the distances between the items they contain.

Step 2: Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.

Step 3: Compute distances (similarities) between the new cluster and each of old clusters. (Single-linkage, complete-linkage and average- linkage)

Step 4: Repeat the steps 2 and 3 until all items are clustered in to a single cluster of size N

Advantages:

- The algorithm can produce an ordering of objects, which may be informative for data display. Smaller clusters are generated, which may helpful in discovery.
- Hierarchical clustering has the advantage that any valid measure of distance can be used.
- The observations themselves are not required; all that is used is a matrix of distances.

Disadvantages:

- They do not scale well; time complexity of at least O (n^2) where n is the total number of objects.
- They can never undo what was done previously.

IV. SUFFIX TREE CLUSTERING ALGORITHM

Suffix Tree Clustering [1, 12] uses the concept of document clustering for clustering the documents. Document clustering is the process of organizing the documents into groups.

Suffix tree clustering is the search result clustering which is used for making the searching efficient. Suffix tree clustering applies on the documents that may be the text document and the web document.

A. *Suffix tree clustering:* [1, 12, 13, 14, and 15]

Suffix Tree Clustering is a hierarchical document clustering. Suffix Tree Clustering is used for extracting the information from large repository of dataset. The dataset is collection of text documents. The purpose of Suffix Tree Clustering is to group the input text according to the identical phrases. A phrase is a sequence of words. Suffix Tree Clustering is used to improve the searching speed as comparison to other clustering algorithms. Suffix Tree uses the tokens to create a suffix tree. The Suffix Tree structure uses the shared suffixes in the documents. And the algorithm uses these suffixes to identify the base clusters.

B. *Suffix Tree Clustering Algorithm:* [1, 12, 17, 18]

There are many document clustering algorithms for efficient search result. They produce the result according to the users query. Suffix tree clustering algorithm is a linear time clustering algorithm (linear in the size of document set) that is based on identifying the phrases. The simplest form of suffix tree clustering is the phrase based clustering. Suffix tree clustering algorithm takes a document as a string. Suffix tree can be easily identifying the documents that would share a common phrase and uses the information for creating the cluster.

Suffix Tree Clustering algorithm uses the suffix tree data structure to represent the data or a tree like data structure for solving problems that involve the strings. It allows the storage of all substring of a given string. Suffix tree is efficiently identify the document set that share common phrases and create the clusters. Before applying the Suffix Tree Clustering on the documents are first cleaned by using Tokenization, Stop words removal, and apply the stemming algorithm.

The Suffix tree data structure is heart of the suffix tree clustering algorithm. A suffix tree is constructed from set of strings. In Suffix Tree Clustering, sentences from documents are inserted to the suffix tree as a word, not as a character.

Documents are marked in internal nodes where the suffixes occur. Internal nodes of the suffix tree represent the phrases that are shared by group of documents. Each internal node represents the base cluster. While traversing the suffix tree clustering algorithm assign a score to every base cluster.

C. Definition of suffix tree [18]: A Suffix Tree ST for an m character string S is rooted directed tree with exactly m leaves numbered 1 to m. Each internal node,

other than the root has at least two children. And each edge is labeled with non-empty substring of S. No two edges out of a node can have edge labels beginning with the same character.

D. Algorithm steps for Suffix Tree Clustering Algorithm [1, 12]

The suffix tree clustering algorithm applies on the documents to fast and efficient searching. Suffix tree clustering produces the better result than the other clustering algorithm. Suffix tree clustering uses the tree like data structure. There are algorithm steps defined for clustering the documents:

Step 1: Collection of documents.
Step 2: Document Cleaning (Preprocessing).
Step 3: Identify the Base clusters.
Step 4: Combining Base clusters.
Step 5: labeling clusters

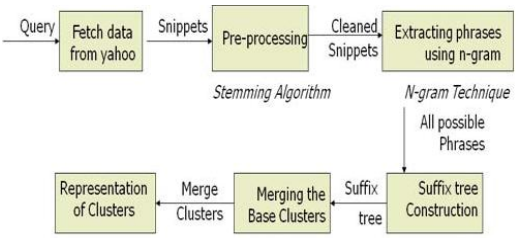

Figure 3: True Common phrase label discovery algorithm

Step 1: Collection of documents:

This step includes the collection of the documents. The collected documents are stored in the dataset. The collected documents are in the form of text documents and the web documents. After that we apply the document cleaning steps on the data set to clean the data.

Step 2: Document Cleaning (Preprocessing):

This is very first step of the STC algorithm that is applied on the collection of the documents that is collected in the first step. In Document cleaning, data is cleaned from the missing values, smoothing noisy data and inconsistencies.

Data cleaning is the preprocessing of the data, through which data is cleaned and processed that is input to the next step to the Suffix Tree Clustering Algorithm. Preprocessing includes the steps such as:

- Tokenization
- Stop- word removal
- Stemming algorithm

Tokenization: In this step of preprocessing, sentences are divided into tokens. Tokenization is the process of identify the word and sentences boundaries in the text. The simplest form of tokenization is the white space character as a word delimiters and selected punctuation mark such as '.','?'and '! '. Each word assigns a token id.

Stop-word removal: many of the words occurring in the text containing no information about its topic. These are called function words, which perform some necessary function in the sentence structure (e.g. link word: and, but are used to form different tenses: will, have etc.) but have no meaning themselves. These words occur so frequently in the sentences. These words should not be considered during processing the text. A list of function word, called stop-list is used and words from this list (called stop-words) are treated as meaningless.

Stemming algorithm: In the stemming procedure all words in the text document are replaced with their respective stem. A stem is a portion of a word that would be left after removing the affixes (suffixes and prefixes). Different form of words can be reduced into one base form by using the stemmer.

Lots of stemmer created for the English language.it is a small piece of code, being just a set of several quite simple rules. The process of stemmer development is easy. There is lot of stemmers available for English language such as: Porter stemmer, Paice stemmer and Lovins Stemmer. For example: connected, connecting, interconnection is transformed into word connect.

Step 3: Identify the Base clusters:

Identification of base cluster can be viewed as creation of phrases for our document collection. This can be done by using the suffix tree data structure. The structure can be constructed in time linear with the size of collection. A suffix tree has two advantages with the mechanism of phrase identification. First, it can find the phrase of any length; second, it is fast and efficient in finding the phrases shared by two or more documents. It is efficient in the sense that most nodes in the suffix tree correspond to maximal phrase cluster. The research based on suffix tree clustering firstly introduced by Zamir. The merging of base clusters can be introduced in the next step.

A suffix tree of string S containing all the suffixes of S. the documents are treated as string of words. The suffixes in the suffix tree containing one or more words. Terms to be used for suffix tree:

- A suffix tree is a rooted tree.
- Each internal node has at least two children.
- Each edge is labeled with non-empty substring of S. the label of a node is defined to be the concatenation of edge label on the path from root to that node.
- No two edges out of same node can have edge labels that begin with the same word.
- For each suffix s of S there exist suffix nodes whose label equal to s.

For example: there are three documents such as:
D1: cat ate cheese.
D2: mouse ate cheese too.
D3: cat ate mouse too

The nodes of the suffix tree are drawn as circles. Each suffix node has two or more boxes attached to it designating the string it originated from. The first number in each box designates the string of origin in our example and the second number designates which

suffix of that string labels that suffix node. Each node in suffix tree represents the group of documents and a phrase that is common to all of them. Each node represents the base cluster.

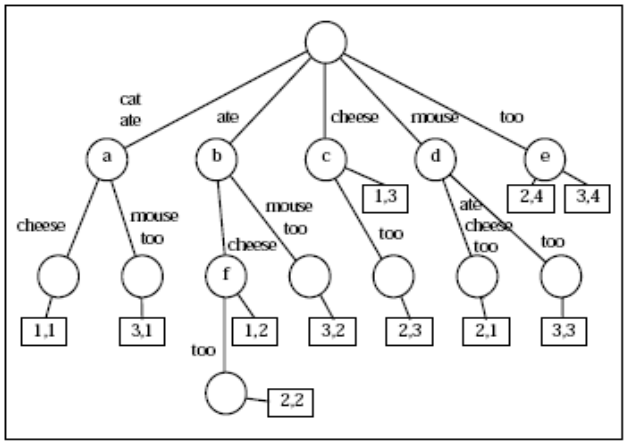

Figure 4: Suffix tree structure for 'cat ate cheese', 'mouse ate cheese' and 'cat ate mouse too'

Each base cluster is assigned a score that is function of the number of documents it contains, and the words that make up its phrases the score s (m) of the base cluster m with phrase P.

The score function calculated for the base clusters, balance the length of phrases, coverage of all candidate clusters (the percent of all collection of document it contain) and the frequency of phrase term in the total collection of documents. A candidate node becomes a base cluster if and only if it exceeds a minimal base cluster score.

$$s(m) = |m| \cdot f(|m_p|) \cdot \sum tfidf(w_i) \quad (6)$$

Where:

S (m) - the score of candidate m

|m| - number of phrase terms

f ($|m_p|$) - phrase length adjustment

tfidf(w_i) - term frequency adjustment

Tfidf is Term Frequency and Inverse Term Frequency measures for assigning weight to terms. The formula which is used to calculate the tfidf:

$$tfidf(w_i, d) = (1 + \log(tf(w_i, d))) \cdot \log(1 + N / df(w_i)) \quad (7)$$

Where:

tf (w_i, d) - number of terms w_i occurred in document d.

N - total number of documents

df (w_i) - number of documents term w_i appear in

Table 1: Base Cluster Table

Node	Phrase	Documents
a	cat ate	1,3
b	ate	1,2,3
c	cheese	1,2
d	mouse	2,3
e	too	2,3
f	ate cheese	1,2

Step 4: Combining (merging) Base clusters:

Phrases are shared by more than one document. The document set of distinct base clusters may overlap and may even be identical. The third step of algorithm of merging the base cluster with high degree of overlap. The similarity measures between base clusters based on overlap of their document set. Base clusters are represented by the base cluster graph. B_m and B_n are two base clusters. $|B_m \cap B_n|$ represent the documents that are common between base clusters. Two nodes are connected if the two base clusters have similarity 1.each cluster the union of the documents of all its base clusters. For example there is a base cluster graph for the above three documents define in the suffix tree:

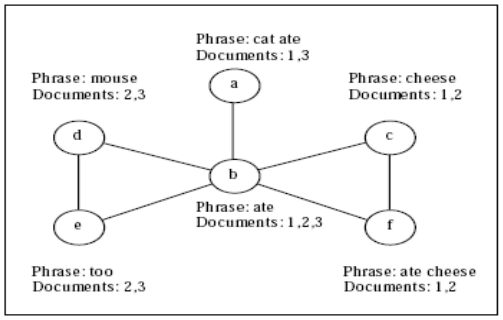

Figure 5: Base Cluster Graph

Base clusters are clustering by using single linkage clustering algorithm. Where there is a minimal similarity is predetermined between base clusters as halting criteria. The clustering algorithm is incremental and order independent. The final clusters are scored and sorted based on the scores of their base clusters and their overlap. The goal of clustering algorithm, in the domain is to group each document with other sharing common topics, but not necessary to partition the collection.

STC algorithm doesn't require specifying the required number of clusters. It requires the specification of threshold used to determine the similarity between base clusters.

Step 5: Labeling clusters:

There are clusters Ci,Cj,…,Cn. Each cluster has set of phrases Pi,Pj,…,Pn and documents Di,Dj,…..,Dn. Each cluster contains the Phrases Pi and Document Di. List of phrase frequency can be maintained by using the formula to calculate the frequency. After merging the final clusters are labeled using highest frequency phrases. This method work well for labeling.

Similarity measures: [3, 5]

Similarity measures calculated by using the binary similarity and cosine similarity etc. cosine similarity can be defined as:

Cosine similarity measures [5]:

Cosine similarity measure is used to calculate the similarity between two documents. There is several ways to compute the similarity between documents. We use the binary similarity and cosine similarity to compute the similarity between the documents. The similarity between the documents is known as the small distance in one cluster. Documents are represented by the vectors where each attribute represent the frequency of word with a particular word occur in the document. The equation which used to calculate the similarity

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (8)$$

Cosine of two vectors can be calculated by using the Euclidean dot product:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta \quad (9)$$

A and B are two vectors of attributes. For text matching the attribute vector of A and B are term frequency vectors of the documents.in case of information retrieval the cosine similarity ranges from 0 to 1 and the term frequency cannot be negative.

Each word in the texts defines a dimension in Euclidean space and the frequency of each word corresponds to the values in the dimension. For example: there are two documents:

Doc1: cat ate cheese.

Doc 2: mouse ate cheese too.

Table 2: Cosine Similarity for documents

Doc/words	cat	ate	cheese	mouse	too
D1	1	1	1	0	0
D2	0	1	1	1	1

Calculation of the similarity by using above equation:

$$\frac{1.0+1.1+1.1+0.1+0.1}{\sqrt{1^2+1^2+1^2+0^2+0^2}\;\sqrt{0^2+1^2+1^2+1^2+1^2}} \simeq 0.54$$

Similarity measures are used to measures the performance and quality of the clustering results. We have to compare the different clustering techniques to get the better performance and quality of cluster while doing searching the information. There are two methods for measure the performance and quality of clusters such as: F-measure and Entropy.

Entropy: the entropy [5] is defined as for a given cluster C_i.

$$Ei = -\Sigma \text{ precision } (i, j) * \log \text{precision } (i, j) \quad (10)$$

Precision (i, j) is the entropy of cluster j in cluster i.

$$\text{Entropy} = \Sigma ((Ni/N)*Ei) \quad (11)$$

If all document in a cluster has same label it means that it has zero entropy otherwise it has positive entropy. Lower entropy means the better quality clusters.

F-measure: F-measure is mainly used for text clustering. It provides the balancing between the precision and recall. They are defined as:

Precision: The percentage of retrieved documents that are relevant to the user query, defined as:

$$Precision = \frac{\{\text{Relevant}\} \cap \{\text{Retrieved}\}}{\{\text{Retrieved}\}} \quad (12)$$

Recall: The percentage of relevant documents that is related to user query. It is defined as:

$$Recall = \frac{\{\text{Relevant}\} \cap \{\text{Retrieved}\}}{\{\text{Relevant}\}} \quad (13)$$

V. CONCLUSIONS

The paper presents the analysis of different clustering techniques such as partitioned clustering and hierarchical clustering. K-means presents the Partitioned clustering and Agglomerative Hierarchical Clustering presents the Hierarchical clustering. It also defines the algorithm steps of these clustering algorithms. This paper also describe algorithm for clustering the web search result, which is known as Suffix Tree Clustering Algorithm. The main steps of STC are to identify the base cluster and merging the base cluster. The work is to be done in this paper is used to help in search engine result easy to browse and quickly find the relevant web information according to user interest. Similarity measures can be used to define the performance of clusters.

VI. FUTURE WORK

The proposed techniques have some advantages and limitations also. The future work is the implementation of tool that is based on the suffix tree clustering algorithm. Another idea is to improve the performance of STC algorithm. Improvements in algorithm, better group content discovery, incremental processing, and cluster label generation. Further improvement in quality of grouping.

REFERENCES

[1] Kale, U. Bharambe, M. Sashi Kumar, "A New Suffix Tree Similarity Measure and Labeling for Web Search Results Clustering", *Proc.* Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09, p.856-861.

[2] (2012).L. B. Ayre, "Data mining for information Professional".

[3] V. M. A. Bai and Dr. D. Manimegalai, "An Analysis of Document Clustering Algorithm", in ICCCCT-10, IEEE 2010, p.402-406.

[4] C.Tsai,T.Liang,J.Ho,C.Yang and M.Chiang, "A Document Clustering Approach for Search Engines",2006 International Conference on System,Man, and Cybernetics October 8-11,2006,Taipei,Talwan,p.1050-1055.

[5] L.Muflikhah and B.Baharudin, "Document Clustering using Concept Space and Cosine Similarity Measurement",2009 International Conference on Computer Technology and Development, 2009 IEEE, p. 58-62.

[6] S.Na,G. yongand L. Xumin, "Research on K-means Clustering Algorithm",Third internation Symposium on intelligent Information Technology and security informatics,2010 IEEE,p. 63-67.

[7] K. A. A. Nazeer, M. P. Sebastian, "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm", Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009, London, U.K.

[8] D. Napoleon and P. G.lakshmi, "An Efficient K-Means Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Points",*Proc*. IEEE 2010,p.42-45.

[9] (2012). "K-Means Clustering Tutorials" http:\\people. revoledu.com\kardi\ tutorial\kMean\.

[10] G. Zhang, Y.Liu, S.Tan, and X.Cheng, "A Novel Method for Hierarchical Clustering of Search Result", 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops.

[11] H.Sun, Z.Liu and L.Kong, "A Document Clustering Method Based on Hierarchical Algorithm with Model Clustering", 22nd International Conference on Advanced Information Networking and Application-Workshops. IEEE 2008, p.1229-1233.

[12] O. Zamir and O. Etzioni, "Web Document Clustering: A Feasibility Demonstration", in *Proc.* the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998, p. 46-54.

[13] H. Chim and X. Deng, "Efficient Phrase-Based Document Similarity for Clustering," IEEE Transaction on Knowledge and Data Engineering, vol. 20, no. September 2008, pp. 1217-1229.

[14] S.osiuski and D.Weiss, "A Concept-Driven Algorithm for Clustering Search Results", IEEE 2005.

[15] H. Wen, G. Luang, and Z.Li , "Clustering Web Search Results Using Semantic Information", in *Proc.* the eighth international conference on machine learning and cybernetics, boarding, 12-15 July 2009 IEEE, p.1504-1059.

[16] D.Zang and Y. Dong, "Semantic, Hierarchical, Online Clustering for Web Search Results".

[17] Rafi, M.Maujood, M.M.Fazal, S.M.Ali, "A Comparision of Two Suffix Tree Based Document Clustering Algorithm", in *Proc*. IEEE 2010NU-FAST, Karachi, Pakistan.

[18] (2011) home page on CS.[Online].Avalable: http://www.cs.gmu.edu/cne/modul e/dau/stat/ clustgalgs/clust5_bdy.html.

[19] J.Han and M.Kamber, "Data Mining Concepts and Techniques", 2nd Edition, 2006 Elsevier.

[20] (2011) Available: http://www.allisons.org.

Pushplata received the Bachelor degree in Computer Science and Engineering from Maharishi Dayanand University Rohtak, India in 2010. She is doing her Master's in Computer Engineering from Maharishi Dayanand University Rohtak (Manav Rachna College of Engineering). Her Research interest is Data Mining (Clustering) including theory and techniques of the data mining.

Mr. Ram Chatterjee received his Master's in Master of Computer Application and M.Tech (Computer Science and Engineering) from CDAC, Noida. He is working as Assistant Professor in Manav Rachna College of Engineering, Computer Science Department, Faridabad - 121004. Haryana, INDIA. His interest area is Data mining and Software Engineering.

